Adversarial methods to reduce simulation bias in neutrino interaction event filtering at liquid argon time projection chambers

M. Babicz,[1,2,*] S. Alonso-Monsalve,[3,2,†] S. Dolan,[2,‡] and K. Terao[4,§]

[1]*Institute of Nuclear Physics Polish Academy of Sciences, PL-31342 Krakow, Poland*
[2]*CERN, The European Organization for Nuclear Research, 1211 Meyrin, Switzerland*
[3]*ETH Zurich, Institute for Particle Physics and Astrophysics, CH-8093 Zurich, Switzerland*
[4]*SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025-7090, USA*

(Received 31 January 2022; accepted 9 May 2022; published 21 June 2022)

For current and future neutrino oscillation experiments using large liquid argon time projection chambers (LAr-TPCs), a key challenge is identifying neutrino interactions from the pervading cosmic-ray background. Rejection of such background is often possible using traditional cut-based selections, but this typically requires the prior use of computationally expensive reconstruction algorithms. This work demonstrates an alternative approach of using a 3D submanifold sparse convolutional network trained on low-level information from the scintillation light signal of interactions inside LAr-TPCs. This technique is applied to example simulations from ICARUS, the far detector of the short baseline neutrino program at Fermilab. The results of the network, show that cosmic background is reduced by up to 76.3% whilst neutrino interaction selection efficiency remains over 98.9%. We further present a way to mitigate potential biases from imperfect input simulations by applying domain adversarial neural networks (DANNs), for which modified simulated samples are introduced to imitate real data and a small portion of them are used for adversarial training. A series of mock-data studies are performed and demonstrate the effectiveness of using DANNs to mitigate biases, showing neutrino interaction selection efficiency performances significantly better than that achieved without the adversarial training.

DOI: 10.1103/PhysRevD.105.112009

I. INTRODUCTION

The current and next generation of neutrino oscillation experiments offer a tantalizing opportunity to explore physics beyond the Standard Model. However, as detectors grow larger and neutrino beams more powerful, a prefiltering of relevant neutrino interaction data becomes increasingly important for experiments to be computationally viable. This is particularly crucial for liquid argon time projection chambers (LAr-TPCs), which are commonly used neutrino detectors (see e.g., [1–5]). LAr-TPCs offer precise spatial and calorimetric measurements based on the electron drift signal from ionization and the scintillation photons from the excitation of argon atoms caused by interacting particles. However, detecting neutrino interactions with this technology becomes challenging due to the significant background of incoming cosmic rays.

For example, at the ICARUS detector of the short baseline neutrino (SBN) experiment, cosmic rays are expected to outnumber neutrino interactions within the booster neutrino beam's (BNB's) spill gate by more than three to one [4]. Even located 1.5 km underground, the DUNE far detectors will experience a comparable rate of cosmic rays and neutrinos [5].

In LAr-TPCs, the ionization electrons, stimulated by propagating charged particles, are drifted by an applied electric field to be collected by TPC anode wires, whilst the emitted LAr scintillation light is recorded by photodetectors, often using photomultiplier tubes (PMT). As such, the TPC records charged particle trajectories as images with a high spatial resolution (∼ mm/pixel) and the photodetectors provide event timing information with nanosecond resolution. The scintillation light signal thereby provides an easily accessible means to classify events requiring little or no processing, which may help distinguish cosmic rays from neutrino interactions before running any reconstruction algorithms.

The rejection of cosmic-ray backgrounds in LAr-TPCs typically starts at the online stage. A trigger to record data is

*Corresponding author.
marta.babicz@cern.ch
†saul.alonso.monsalve@cern.ch
‡stephen.joseph.dolan@cern.ch
§kterao@slac.stanford.edu

issued only if a fast signal from the photodetectors is observed in the beam spill window, the time window during which neutrino signal is expected. However, it does not prevent the selection of background events caused by an accidental coincidence of a beam spill window with incident cosmic rays. More sophisticated filtering may be achieved through multidimensional analyses, as discrimination power can be found by analyzing which PMTs received light, at what time, and for how long relative to all other PMTs inside the LAr-TPC.

Machine-learning methods, capable of analyzing such high-dimensional information, are therefore excellently suited to classifying events using the available PMT information. In particular, when detector data is represented as images, the use of convolutional neural networks (CNNs) [6,7] are especially effective. The use of CNNs for event classification is well established across the field of neutrino physics [8–10]. Although CNNs are broadly used in the neutrino community, images of neutrino interactions are typically very sparse such that most of the pixels have empty values which can render some standard methods ineffective. A straightforward solution to this issue is to use submanifold sparse convolutional networks (SSCNs) [11], a variation of standard CNNs that use a new sparse convolutional operator to efficiently handle sparse inputs, with already a remarkable number of successful applications in neutrino physics [12,13]. All the neural network architectures shown in this paper belong to the class of SSCNs. Even though the customization of the network architecture is not the main purpose of this paper, further optimization of other architectures, such as graph neural networks [14,15], may have been able to provide similar results.

CNNs[1] can be optimized to discriminate signal images against backgrounds through a supervised training process. This is often done using simulated images (e.g., simulated neutrino and cosmic images) where the true labels are available. However, when this model is applied to images from the real detector, its performance is typically worse than what is observed on simulated images because of discrepancies between two data domains (i.e., physics of the real world vs simulation) due to imperfect simulation. To address this problem, the CNN classifier may rely on domain adaptation (DA) techniques [16,17] so that the classifier learned from the training domain (i.e., simulated data) can also be applied to the testing domain (i.e., eventual experimental data). This DA can be achieved through the application of domain adversarial neural networks (DANNs) [18], in which the detector data is used in an unsupervised (or semisupervised) manner to prevent the CNN exploiting features that differ between data and simulation. DANNs were first used in neutrino physics by the MINERνA experiment, where the bias of a deep-learning-based neutrino vertex identification method was mitigated using these techniques [19]. In this paper, we present the first application of DANN for a CNN as an event classifier for a LAr-TPC to discriminate neutrino signal against cosmic backgrounds.

To test the effectiveness of CNNs and DANNs at distinguishing cosmic-ray backgrounds from neutrino interactions using only information from the scintillation signal, we consider the ICARUS detector of the SBN program [4] as a case study. ICARUS is currently the world's largest LAr-TPC employed in neutrino physics and operates close to the surface, and so is subject to a particularly challenging cosmic-ray background rejection. The details of the ICARUS detector and simulation are summarized in Sec. II. The CNN approach to event filtering is detailed and demonstrated in Sec. III. The application of DANNs to reduce the CNN sensitivity to input simulation dependence, and a method of using mock-data studies to test their effectiveness, is then described and applied in Sec. IV. Finally, the results and the main conclusions of this work are presented in Sec. V.

II. EVENT FILTERING AT THE ICARUS DETECTOR

The ICARUS detector [20] is a 760-ton LAr-TPC, serving as the far detector of the SBN program [4], positioned 600 m away from the booster neutrino beam at FNAL. The detector consists of two identical adjacent modules, each housing two TPCs separated by a common cathode used to generate the electric field that directs the argon ionization signal to the anode. The prompt (order of nanosecond) LAr scintillation light signal from charged particles propagating within ICARUS is readout by 360 8 inch PMTs [21] arranged on the walls of the TPCs, placed as shown in Fig. 1. The PMT system provides the means to trigger the readout of signals within the 1.6 μs beam spill windows whilst also enabling fast spatial localization of neutrino beam associated events. The placement, performance, and timing resolution of the PMTs are expected to allow the localization of the associated charged particle

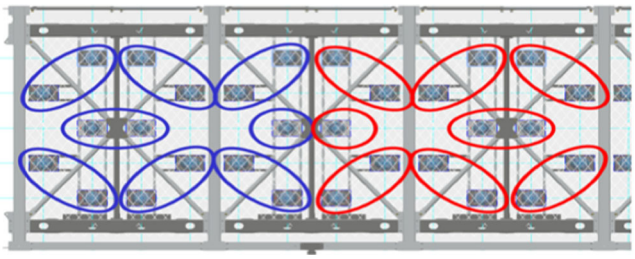

FIG. 1. A schematic view of a one wall segment of ICARUS TPCs showing how the PMTs are distributed. The PMTs are arranged in groups of 15 that are connected to the same digitizer board. The hollow ellipses show the pairing system of adjacent PMTs.

[1]We refer to CNNs and SSCNs indistinguishably in the rest of this section.

track with accuracy better than 1 m [21–23]. The signals of all PMTs are continuously read out, where pairs of adjacent PMTs are typically used together within the ICARUS trigger system (the pairing scheme is shown in Fig. 1). For each beam spill window, the ICARUS trigger system can assess which PMTs have a signal exceeding a predefined threshold, at what time that signal is recorded with respect to the start of the beam window and how many times the PMT recorded an opening above the threshold. Note that a new opening is counted every 0.16 μs that the PMT signal remains above a preset threshold and so the number of openings acts as a discretized measurement of the time over the threshold, which itself is highly correlated with the signal amplitude.

The BNB delivers neutrinos in a bunch at the rate up to 5 Hz. The accelerator complex informs the trigger system of the arrival time of each bunch, and hence the timing of the beam spill window, during which the detector may issue a trigger if there is a signal in photodetectors above the threshold. This results in about 5% of beam spills being recorded. Despite a large reduction factor, the recorded events are still dominated by cosmic-ray backgrounds due to an accidental coincidence of the beam-window and optical signals produced by cosmic rays. Our goal is to further reduce the cosmic-ray backgrounds via CNN-based fast event filter that only requires optical data.

Given the possibility of better separation of neutrino interactions from cosmic rays after full reconstruction of the TPC signal, the primary aim of the low-level event filter is to reduce the vast majority of cosmic rays while not compromising neutrino selection efficiency. In this way, the amount of data that needs to be processed for higher-level analyses is greatly reduced while avoiding the risk of losing the neutrino signal. In order to allow the possibility of online event filtering, only the information available to the ICARUS event trigger is used. This means that information from each PMT pair is concatenated where the earliest opening time from each pair is stored alongside the total number of openings across both PMTs within the pair.

A. Simulation

The cosmic-ray particles impinging the ICARUS detector are generated with CORSIKA event generator [24]. These particles are then propagated through the ICARUS detector and the surrounding material using GEANT4 [25] implementation in LArSoft [26]. Scintillation photons are then propagated to the PMTs using a parametrized model based on precalculated tables (also derived from GEANT4). A further parametrized PMT readout model, constrained from test-beam data, is then used to simulate the digitized datalike signal from the detector.

The incoming flux of neutrinos is modeled using a GEANT4-based simulation of the BNB beamline [4,27] whilst their interactions with the nuclei (and electrons) within ICARUS are modeled using GENIE version 3 [28]. The particle propagation and detector response are simulated identically to the case of cosmic rays.

For this study, we simulate 396,200 PMT readout windows (events) containing cosmic rays and 120,000 containing a single neutrino interaction. For this work, only one of ICARUS' two cryostats is considered. Whilst it is possible to have a PMT readout window containing cosmic rays and a neutrino interaction or multiple neutrino interactions, this is not particularly common, and such details are beyond the scope of this study. The number of events used in this work is reduced to those that passed the ICARUS trigger conditions, resulting in 114,589 neutrino and 46,115 cosmic events.

III. CNN EVENT FILTER

The goal of the CNN is to classify whether events are from neutrino or cosmic-ray interactions. To train the CNN, the simulated PMT data is presented as 3D images, where the position of each PMT pair, alongside its opening time and a number of openings, are stored. The image voxel size is chosen to be 40 cm, which is the maximum distance such that two PMT pairs do not appear within the same voxel. An example of the image provided as input, divided into two subimages representing the two PMT pair observable for better visualization, is shown in Fig. 2. Each event contains one image, which expresses both the opening time and number of openings for each PMT pair, of which 80% are used for training, 10% for validation, and the remaining 10% for testing. Each image in the training sample is labeled as a cosmic-ray or neutrino event.

The main feature of CNNs is that they learn a series of filters (using convolutions), applied in sequence to extract increasingly powerful and abstract features that allow the CNN to learn a mapping between input images and target labels. Once the CNN is trained, it can be applied to new images to make accurate predictions on unseen examples during the training. The designed CNN architecture is depicted in Fig. 3. As introduced in Sec. I, it is based on 3D submanifold sparse convolutions [11] to deal with the sparse images used in this work. The CNN is trained[2] for 50 epochs[3]—with a cross-entropy loss—using PYTHON3.6.9 and PyTorch2.1.0 [29], as well as the MinkowskiEngine package, version 0.5.4 [30], on an NVIDIA Tesla V100 GPUs. Stochastic gradient descent (SGD) is used as the optimizer, with a minibatch size of 32 events, a learning rate of 0.1 (divided by 10 when the error plateaus, as suggested in [31]), a weight decay of 0.0001, and a momentum of 0.9.[4]

[2]The rest of this paragraph applies to all the neural networks analyzed in this article.

[3]Epoch: one forward pass and one backward pass of all the training examples. In other words, an epoch is one pass over the entire dataset.

[4]See Ref. [32] for a description of optimizers and associated terminology.

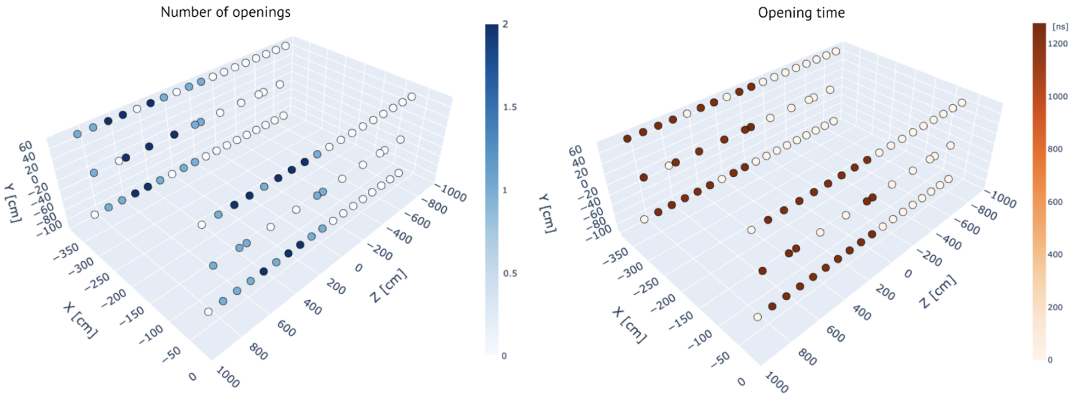

FIG. 2. Example images of one ICARUS cryostat used as an input to the CNN. Each dot represents a PMT pair position (taken as the pair's barycenter) which are distributed across the walls of the TPCs. The color of the dots represents the number of openings (left) or the opening time (right).

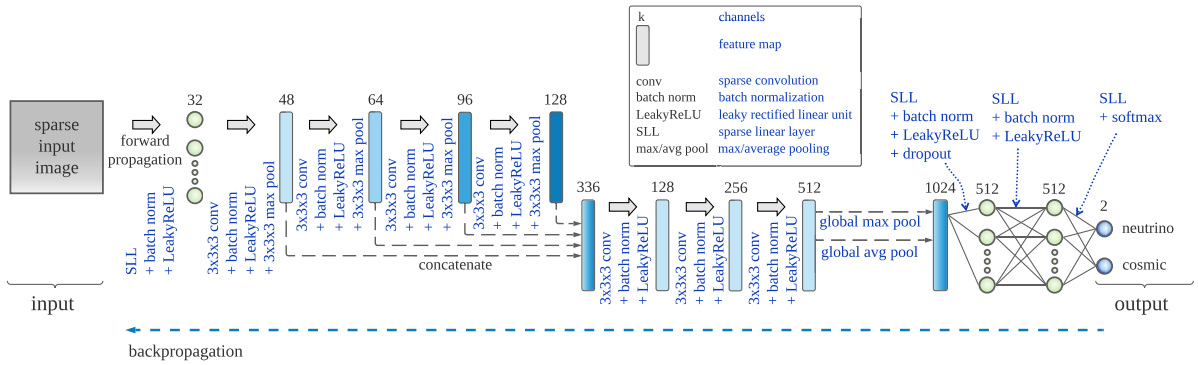

FIG. 3. The sparse convolutional network architecture used for this analysis. It was developed using the MinkowskiEngine package [30] to handle sparse inputs more efficiently.

The model weights used for the analysis correspond to those at the epoch that maximizes the overall accuracy on the validation set. Figure 4 shows the evolution of the cross-entropy loss and the accuracy (i.e., the proportion of events correctly classified) within the validation and training samples as a function of the epoch. Since these metrics appear almost identical between the two samples, there is no sign of overtraining. The accuracy curves look flat due to the nearly identical shape of ∼19% of the neutrinos and cosmics, making the separation task exceptionally complicated for those events. Moreover, the loss function converged over time, and the score distributions improved accordingly. Besides, the same behavior was reported for different configurations of optimizers and learning rates, discarding any problems with the training. The final model configuration is that obtained at epoch 33, which maximizes the accuracy of the validation sample.

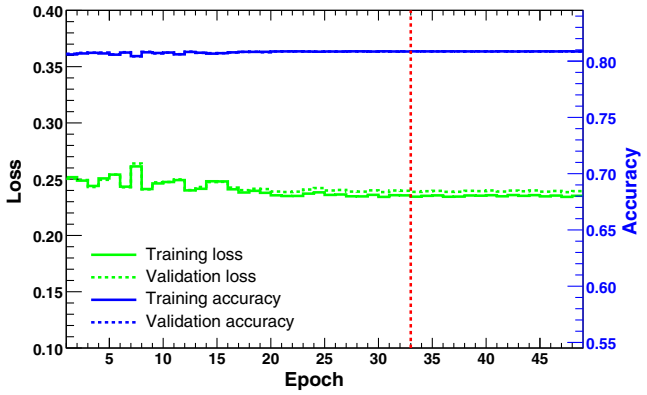

FIG. 4. The loss and accuracy per epoch of the training and validation samples. The red vertical line highlights the epoch that maximizes the accuracy on the validation sample and is used as for subsequent analysis.

A. Performance

Once trained, the output of the CNN is a continuous score for each event between 0 (neutrinolike) and 1 (cosmiclike). The distribution of CNN scores for each true event type in the test sample is shown in Fig. 5. If a selection of neutrino events is made by cutting at a CNN

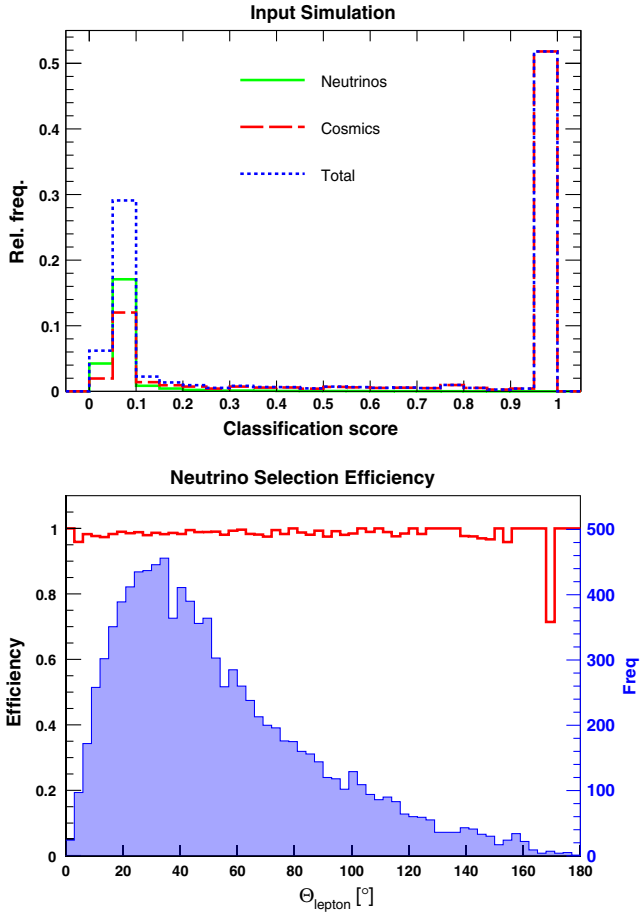

FIG. 5. The trained CNN's classification of events in the test sample (top). Note that the relative normalization of cosmic and neutrino events is fixed to approximate what would be expected in data. The distribution of outgoing lepton angle (blue) with respect to the incoming neutrino from GENIE is shown alongside the neutrino selection efficiency of the CNN (red) following a cut at a classification score of 0.5 (bottom).

score of 0.5, a 98.9% selection efficiency is maintained whilst 76.3% of cosmic-ray backgrounds are rejected. The charged-current selection efficiency is found to be flat (i.e., unbiased by kinematics) in various tested observables. An example of an outgoing lepton angle is shown in Fig. 5.

IV. REDUCING MODEL-DEPENDENCE WITH DANN-BASED TRAINING

Whilst the CNN presented in Sec. III A shows excellent performance, the results assume perfect modeling of the neutrino and cosmic-ray events, the particle propagation and the detector response. If the CNN is trained with events that do not suitably represent what is in the real data, then the test sample's performance will not be reliable. Modern deep neural networks consist of millions (and sometimes billions) of parameters and have a strong representative capability with which they may exploit every detailed feature present in the simulation, including those that may not be present in data as well as others that may not follow the true physics model behind real data. Thus, it is not easy to ever be sure that the pertinent aspects of the events are well modeled. To alleviate this issue, adversarial training methods can be employed to prevent neural networks from exploiting features that are only present in one of two domains. As a result, the performance can be made consistent in both domains. In this analysis, we show that it is possible to mitigate challenges associated with domain discrepancies through the application of DANNs.

In DANNs, the neural network model is trained on examples from two domains: (a) the source domain, which consists of labeled simulated data; and (b) the target domain, which consists of unlabeled true experimental data. The goal is to learn a discriminator from the labeled source domain examples and use the unlabeled target domain examples to ensure the discriminator relies on only domain-invariant features to perform the predictions. Regarding the implementation of the neural network, the classifier architecture remains identical, and it can be seen as the combination of a feature extractor (i.e., the bulk of the CNN, in our case) and a label predictor [i.e., the sparse linear layer(s) at the end]. However, this alternative neural network has a second path, which connects the output of the feature extractor through a gradient reversal layer with a few linear layers that form a domain classifier. The gradient reversal layer performs an identity transformation during the forward propagation process but multiplies the gradient by a negative constant during the back-propagation, guaranteeing that the parameters learnt by the feature extractor are made similar for the source and target distributions. In other words, with this approach, the features learnt by this model are simultaneously discriminative (thanks to the label predictor), and domain-invariant (thanks to the domain classifier). This behavior is shown in Fig. 6. Furthermore, if some events from the target distribution are labeled (e.g., experimental data cosmic rays produced without a neutrino beam), those events might be used for the feature extractor learning too, making the domain adaptation semisupervised, in contrast to the unsupervised case where all the events from the target distribution are unlabeled.

In order to test the effectiveness of DANNs as a method of reducing simulation dependence, we perform a series of mock-data studies. For these studies, statistically independent simulations of events (from neutrinos and cosmic rays) are produced before being modified to simulate possible mismodeling bias. Since the coarse PMT information used in this analysis is likely not sensitive to the exact details of the neutrino interaction or cosmic-ray production, we focus primarily on applying distortions to the simulated detector response. The details of the mock data are as follows:

"Global noise" data: in this mock data, noise, which is uncorrelated with the event content, is randomly added to each PMT with some prespecified 'global' probability that

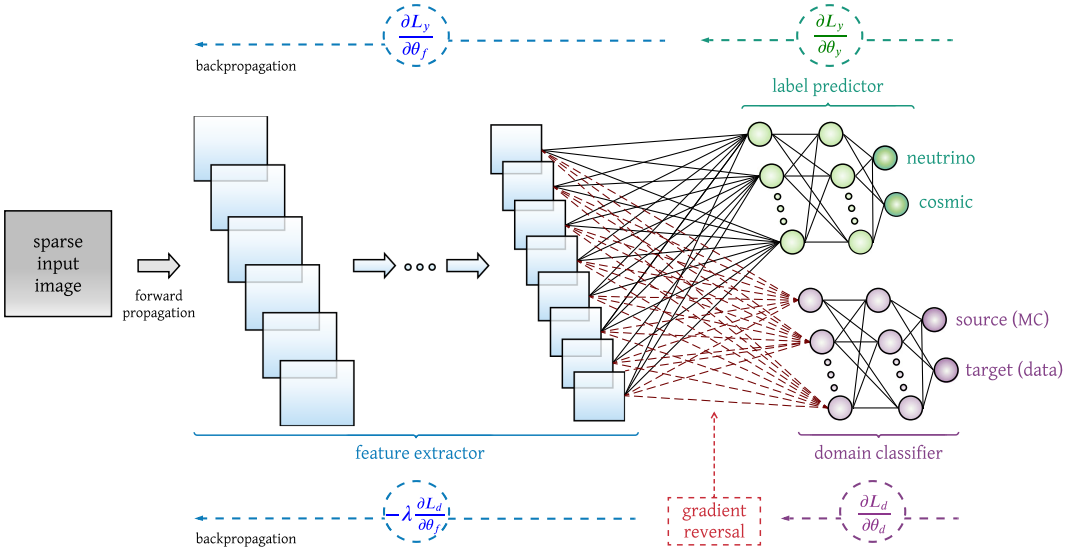

FIG. 6. Domain-adversarial neural network architecture. The feature extractor (blue) and the label predictor (green) form the standard neural network classifier shown in Fig. 3. The domain classifier (purple) provides the domain adaptation part since it is connected to the feature extractor through a gradient reversal layer, allowing the alignment of feature distributions across the source and target domains. Figure adapted from [18].

is common to all PMTs. The global noise probabilities considered are 2%, 5%, and 10%. The timing of the noise is modeled as uniform distribution. If noise is simulated to arrive before a PMT is opened by a simulated signal, the opening time recorded by that PMT is overwritten by time of the noise.

"Local noise" data: similarly to the global noise data, this mock dataset considers the addition of random noise to each PMT but where the probability of producing noise is different for every PMT. Noise probabilities per event for each PMT were generated randomly using a uniform distribution between 0 and either 2%, 5%, or 10%.

For each mock-data study, the DANN is trained as described in Sec. III but with an addition of 22,778 cosmic-ray-mock-data events and 94,306 neutrino-mock-data events (40% train, 10% validation, 50% test),[5] which are labeled by the domain (i.e., mock data or original simulation), all the simulated events are labeled by event type (i.e., cosmic or neutrino), and we consider three scenarios for the event type of mock-data events: (1) none of the events are labeled by event type (unsupervised domain adaptation), (2) 10% of the cosmic-ray-mock-data events are labeled (semisupervised domain adaptation), and (3) 50% of the cosmic-ray-mock-data events are labeled (semisupervised domain adaptation). This method could equally be applied to real data instead of mock data, using run periods with no neutrino beam to label the cosmic-ray events.

Both the originally trained CNN (as described in Sec. III) and the newly trained DANNs are used to attempt to classify events from the original sample and from the new mock-data sample. An example of the classification scores for each model applied to the original and mock datasets is shown for two mock-data studies in Fig. 7. A summary of the neutrino selection efficiency and the background rejection performance for the nominal simulation as well as for each mock dataset is shown in Table I. The presented numbers are provided for a selection cut set to 0.5 of the network classification score in all cases. It should be noted that there is additional freedom to optimize performance as desired by tuning the cut value applied. For example, it is found that for the case of mock data with global noise of 10%, the results using an unsupervised domain adaptation improve with respect to the original model by ∼14% for neutrino selection efficiency and ∼1% for cosmic-background rejection by setting a cut at 0.25. However, with the cut at 0.5, the neutrino selection efficiency improves dramatically (by ∼22%), but the background rejection performance decreases (by ∼8%). An alternative assessment of the CNN and DANN performance where the cut is varied to keep the background rejection factor constant is presented in Appendix.

These results show that, without the adversarial training, the original CNN can reject a sizeable portion of neutrino interactions in the mock data. However, once the adversarial training is used, the network is able to mitigate the bias

[5]Following the suggestion in [19], we do not use all the available mock-data events for training. Moreover, the large test set size provides enough statistics for the analysis.

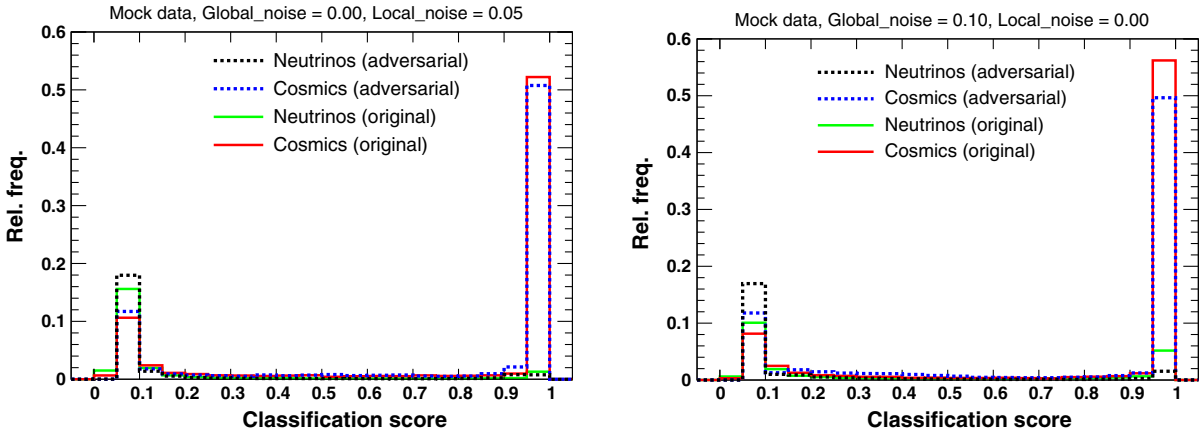

FIG. 7. Results of the classification score of the nominal and adversarially trained CNN/DANN model applied to the original and local-noise mock data using a 5% noise spread (left) and the global-noise mock data using a 10% noise spread (right). None of the cosmic-ray-mock-data events are labeled by event type.

and maintain a very high neutrino selection efficiency (the main goal of the filter) whilst continuing to achieve a significant rejection of cosmic-ray backgrounds. This occurs for the unsupervised and the semisupervised domain adaptations, and it is more visible for larger noises (i.e., 10% of global noise and 10% of local noise). It can equally be observed that the use of a DANN does not degrade the performance of the baseline CNN when applied to the nominal simulation. This demonstrates that even if the simulation very well describes the data, the use of a DANN over a CNN is not expected to degrade the performance.

Concerning the unsupervised domain adaptation compared to the semisupervised cases, we find a small but non-negligible improvement in the rejection of mock-data cosmic-ray backgrounds for some mock-data studies, while

TABLE I. Efficiency ($\mathcal{E}_\nu$) and proportion of rejected cosmic-ray background events ($\mathcal{R}_{\cos}$) using the original and adversarially trained CNN/DANN to classify events in the nominal simulations and mock-data studies. Table II details the model-type nomenclature.

						Domain adaptation					
						Unsupervised		Semisupervised 10%		Semisupervised 50%	
	Dataset		Model type	$\mathcal{E}_\nu$[%]	$\mathcal{R}_{\cos}$[%]	$\mathcal{E}_\nu$[%]	$\mathcal{R}_{\cos}$[%]	$\mathcal{E}_\nu$[%]	$\mathcal{R}_{\cos}$[%]	$\mathcal{E}_\nu$[%]	$\mathcal{R}_{\cos}$[%]
Mock data	Nominal		CNN	98.9	76.3	⋯		⋯		⋯	
			DANNG2	⋯		97.2	77.2	99.0	74.8	97.9	77.2
			DANNG5	⋯		98.3	76.8	98.0	76.6	97.9	76.9
			DANNG10	⋯		98.7	75.8	98.1	76.4	98.5	76.4
			DANNL2	⋯		98.2	76.1	98.1	75.8	98.7	76.1
			DANNL5	⋯		98.8	75.9	98.7	76.0	98.8	76.2
			DANNL10	⋯		98.7	75.2	98.1	76.8	98.0	76.8
	Global noise	2%	CNN	91.7	74.8	⋯		⋯		⋯	
			DANNG2	⋯		92.7	76.5	92.7	75.1	87.2	78.7
		5%	CNN	81.0	75.8	⋯		⋯		⋯	
			DANNG5	⋯		89.7	72.1	89.9	76.6	84.1	78.7
		10%	CNN	66.4	79.0	⋯		⋯		⋯	
			DANNG10	⋯		88.8	71.0	87.2	69.8	88.1	75.9
	Local noise	2%	CNN	95.5	75.1	⋯		⋯		⋯	
			DANNL2	⋯		97.6	74.2	98.2	74.8	96.5	76.6
		5%	CNN	90.2	74.8	⋯		⋯		⋯	
			DANNL5	⋯		89.9	75.0	90.2	75.9	90.3	78.9
		10%	CNN	81.9	75.7	⋯		⋯		⋯	
			DANNL10	⋯		90.2	73.3	88.7	77.9	88.4	78.9

TABLE II. Networks legend.

Network name	Description
CNN	Original neural network trained on the nominal simulation.
DANNG2	Adversarial network trained on nominal simulation + mock data (global noise = 2%).
DANNG5	Adversarial network trained on nominal simulation + mock data (global noise = 5%).
DANNG10	Adversarial network trained on nominal simulation + mock data (global noise = 10%).
DANNL2	Adversarial network trained on nominal simulation + mock data (local noise = 2%).
DANNL5	Adversarial network trained on nominal simulation + mock data (local noise = 5%).
DANNL10	Adversarial network trained on nominal simulation + mock data (local noise = 10%).

suffering a slight reduction in mock-data neutrino selection efficiency. This behavior is expected since, for the semi-supervised cases, the models have more labeled cosmic events to learn to reject from (especially for the case where 50% of the mock-data cosmic rays are labeled). It is possible that labeling a larger portion of the training sample may allow improved performance from the semisupervision (recall that the majority of events in the training sample are neutrino interactions, which cannot be labeled in real data).

V. CONCLUSION

The studies presented in this paper demonstrate that easily accessible information from LAr-TPC experiment's light detection systems, which requires very little processing, may be used to effectively separate neutrino from cosmic-ray induced signals within a neutrino beam spill. The use of a specially adapted CNN ensures that the majority of cosmic-ray interactions can be filtered out without the rejection of almost any neutrino induced interactions.

Whilst the use of a CNN trained on simulated event samples is susceptible to bias due to mismodeling, potentially causing the inadvertent rejection of neutrino events, it is demonstrated that adversarial training via a DANN can mitigate the loss of efficiency at the cost of some reduced background rejection. It is further shown that in some cases, the background rejection performance may be improved through semisupervised domain adaptation of the DANN using labeled real cosmic ray events.

Overall the techniques presented in this manuscript demonstrate a method for providing a significant rejection of cosmic-ray events without the need for computationally expensive reconstruction algorithms. These methods are shown to be effective when applied to simulations from the ICARUS experiment, but are easily adaptable and could likely achieve similar success if applied to other LAr-TPC experiments.

ACKNOWLEDGMENTS

The authors would like to thank the ICARUS collaboration for supporting this work. Particular thanks is given to François Drieslma for providing detailed comments on a draft version of this manuscript. The work of M. B. resulted from the implementation of the research Project No. 2019/33/N/ST2/02874 funded by the National Science Centre, Poland.

APPENDIX: COMPLEMENTARY RESULTS

As discussed in Sec. IV, the performance of the CNN and DANNs' application to the mock datasets may be demonstrated in alternative ways to as presented in Table I, which shows the efficiency and background rejection achieved for a fixed cut in the networks' output classification score distribution. As such, Table III instead changes the cut such that the background rejection remains fixed (at 75%) so that the efficiencies can be more directly compared. The conclusions remain unchanged from those presented in Sec. IV; the performance improvement offered by the adversarial training of the DANNs is substantial with respect to naively applying CNN trained only on the input simulation. The improvement can be seen to be stronger for more extreme fake data studies. Labeling some proportion of the cosmic ray events in the mock-data samples to provide a semi-supervised of the DANNs can offer a small additional improvement in some cases.

TABLE III. Efficiency ($\mathcal{E}_\nu$) using the original and adversarially trained CNN/DANN to classify events in the nominal simulations and mock-data studies. In contrast to what is shown in Table I and for a better comparison of the efficiencies, the classification score cuts were tuned so that the proportion of rejected cosmic-ray background events for each model is always 75%. Table II details the model-type nomenclature.

Dataset			Model type	$\mathcal{E}_\nu$[%]	Domain adaptation		
					Unsupervised $\mathcal{E}_\nu$[%]	Semisupervised 10% $\mathcal{E}_\nu$[%]	Semisupervised 50% $\mathcal{E}_\nu$[%]
Mock data	Nominal		CNN	99.2	···	···	···
			DANNG2	···	98.3	98.9	98.8
			DANNG5	···	99.0	98.7	98.9
			DANNG10	···	99.0	98.6	98.7
			DANNL2	···	98.6	98.3	99.1
			DANNL5	···	99.1	98.9	99.0
			DANNL10	···	98.7	98.8	98.8
	Global noise	2%	CNN	91.5	···	···	···
			DANNG2	···	93.6	92.7	89.6
		5%	CNN	81.7	···	···	···
			DANNG5	···	88.0	90.8	87.3
		10%	CNN	74.7	···	···	···
			DANNG10	···	86.0	84.5	88.6
	Local noise	2%	CNN	95.4	···	···	···
			DANNL2	···	97.1	97.9	97.1
		5%	CNN	89.9	···	···	···
			DANNL5	···	89.9	90.6	93.7
		10%	CNN	82.6	···	···	···
			DANNL10	···	89.1	90.5	91.3

[1] C. Rubbia, The liquid argon time projection chamber: A new concept for neutrino detectors, Reports No. CERN-EP-INT-77-08, CERN-EP-77-08, 1977, http://cdsweb.cern.ch/record/117852/files/CERN-EP-INT-77-8.pdf.

[2] W. J. Willis and V. Radeka, Liquid argon ionization chambers as total absorption detectors, Nucl. Instrum. Methods **120**, 221 (1974).

[3] R. Acciarri *et al.*, First observation of low energy electron neutrinos in a liquid argon time projection chamber, Phys. Rev. D **95**, 072005 (2017).

[4] M. Antonello *et al.*, A proposal for a three detector short-baseline neutrino oscillation program in the Fermilab booster neutrino beam, arXiv:1503.01520.

[5] B. Abi *et al.*, Deep underground neutrino experiment (DUNE), far detector technical design report, volume I introduction to DUNE, J. Instrum. **15**, T08008 (2020).

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. **1**, 541 (1989).

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE **86**, 2278 (1998).

[8] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. D. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, and P. Vahle, A convolutional neural network neutrino event classifier, J. Instrum. **11**, P09001 (2016).

[9] B. Abi *et al.*, Neutrino interaction classification with a convolutional neural network in the DUNE far detector, Phys. Rev. D **102**, 092003 (2020).

[10] C. Adams *et al.*, Deep neural network for pixel-level electromagnetic particle identification in the MicroBooNE liquid argon time projection chamber, Phys. Rev. D **99**, 092001 (2019).

[11] B. Graham and L. van der Maaten, Submanifold sparse convolutional networks, arXiv:1706.01307.

[12] C. Adams *et al.*, Deep neural network for pixel-level electromagnetic particle identification in the MicroBooNE liquid argon time projection chamber, Phys. Rev. D **99**, 092001 (2019).

[13] L. Dominé and K. Terao, Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data, Phys. Rev. D **102**, 012005 (2020).

[14] A. Sperduti and A. Starita, Supervised neural networks for the classification of structures, IEEE Trans. Neural Networks **8**, 714 (1997).

[15] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, Graph neural networks: A review of methods and applications, arXiv:1812.08434.

[16] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, A theory of learning from different domains, Mach. Learn. **79**, 151 (2010).
[17] I. Redko *et al.*, *Advances in Domain Adaptation Theory* (Elsevier, New York, 2019).
[18] Y. Ganin *et al.*, Domain-adversarial training of neural networks, J. Mach. Learn. Res. **17**, 1 (2016).
[19] G. N. Perdue, A. Ghosh, M. Wospakrik, F. Akbar, D. A. Andrade, M. Ascencio, L. Bellantoni, A. Bercellie, M. Betancourt, G. F. R. Caceres Vera *et al.*, Reducing model bias in a deep learning classifier using domain adversarial neural networks in the minerva experiment, J. Instrum. **13**, P11020 (2018).
[20] S. Amerio *et al.*, Design, construction and tests of the ICARUS T600 detector, Nucl. Instrum. Methods Phys. Res., Sect. A **527**, 329 (2004).
[21] M. Babicz *et al.*, Test and characterization of 400 Hamamatsu R5912-MOD photomultiplier tubes for the ICARUS T600 detector, J. Instrum. **13**, P10030 (2018).
[22] B. Ali-Mohammadzadeh *et al.*, Design and implementation of the new scintillation light detection system of ICARUS T600, J. Instrum. **15**, T10007 (2020).
[23] M. Babicz *et al.*, A particle detector that exploits liquid argon scintillation light, Nucl. Instrum. Methods Phys. Res., Sect. A **958**, 162421 (2020).
[24] D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, and T. Thouw, CORSIKA: A Monte Carlo code to simulate extensive air showers, Report No. FZKA-6019, 1998, https://publikationen.bibliothek.kit.edu/270043064.
[25] S. Agostinelli *et al.*, GEANT4–a simulation toolkit, Nucl. Instrum. Methods Phys. Res., Sect. A **506**, 250 (2003).
[26] E. L. Snider and G. Petrillo, LArSoft: Toolkit for simulation, reconstruction and analysis of liquid argon TPC neutrino detectors, J. Phys. Conf. Ser. **898**, 042057 (2017).
[27] A. A. Aguilar-Arevalo *et al.*, The neutrino flux prediction at MiniBooNE, Phys. Rev. D **79**, 072002 (2009).
[28] C. Andreopoulos *et al.*, The GENIE neutrino Monte Carlo generator, Nucl. Instrum. Methods Phys. Res., Sect. A **614**, 87 (2010).
[29] M. Abadi *et al.*, TensorFlow: A system for large-scale machine learning, in *OSDI* (USENIX Association, Savannah, GA, 2016), Vol. 16, pp. 265–283.
[30] C. Choy, J. Gwak, and S. Savarese, 4D spatio-temporal convnets: Minkowski convolutional neural networks, arXiv:1904.08755.
[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).